

**PROCEDURES AND TEMPLATES FOR
CONFIGURATION MANAGEMENT
AND
QUALITY ASSURANCE**

Release Date: December 11, 2000

Contract Number: GS-35F-4919H
Order Number: T0000AJ3739

CLIN #011-1

Prepared for:

U.S. DEPARTMENT OF EDUCATION
OFFICE OF STUDENT FINANCIAL ASSISTANCE (SFA)
400 Maryland Avenue, SW
Washington, DC 20202

Prepared by:

CTGi
Suite 250
10461 White Granite Drive
Oakton, VA 22124



This page left intentionally left blank

FOREWORD

CTG, Incorporated (CTGi) would like to thank the following personnel whose dedication and involvement made the development and completion of this document possible.

Kelly Conboy

Glenn Foster

Candace Jones

Frank Majewski

Annette Mazie

Antony Mosley

Paul Stocks

Sandra Stocks
Director Of Operations,
Washington Metropolitan Area
CTGi

This page intentionally left blank

EXECUTIVE SUMMARY

This document, prepared for the United States (U.S.) Department of Education, Office of Student Financial Assistance (SFA), develops a standardized procedure for the Configuration Management (CM) and Quality Assurance (QA) during System Integration and Testing (SI&T) phases.

This document, along with those listed below, will be integrated into the U.S. Department of Education, SFA, SI&T Process Handbook, which will then become integrated into the overall U.S. Department of Education, SFA Modernization Technology Handbook. The remaining documents that will comprise the U.S. Department of Education, SFA, SI&T Process Handbook are:

- System Integration and Testing Standards
- Test Performance Measurements
- Procedures and Templates for Creating Test Conditions, Test Scenarios, and Testing Data
- Procedures and Templates for Test Execution, Test Evaluation, and Error Correction
- Procedures For Using Testing Tools

Each of the above listed procedures, templates, and guide was prepared for delivery as a separate document.

All SI&T guidelines and procedures are focused on supporting systems and projects used in the development and execution of a comprehensive integration and testing program. To this end, this document contains information on understanding issues related to test planning roles, responsibilities of the organization, documentation requirements, and required testing practices. The standards and procedures used in the support of this document are reflective of industry best practices, practices of other federal government agencies and various governing standards and literature regarding CM and QA procedures during the integration and testing process.

This page intentionally left blank

TABLE OF CONTENTS

Section	Page
1. INTRODUCTION	1-1
1.1 Background	1-1
1.2 Objective	1-1
1.3 Applicability	1-1
1.4 Assumptions.....	1-1
1.5 Document Organization	1-2
2. CONFIGURATION MANAGEMENT	2-1
2.1 Configuration Management Overview	2-1
2.1.1 Participants.....	2-2
2.1.2 Baseline Control.....	2-2
2.1.3 Version Control and Identification	2-3
2.1.4 Change Control	2-4
2.1.5 Review Participation	2-5
3. CONFIGURATION MANAGEMENT FOR TEST ANALYSIS AND PLANNING	3-1
3.1 System Test Plan Creation	3-1
3.1.1 System Test Plan Entrance Criteria	3-1
3.1.2 System Test Plan Input	3-1
3.1.3 System Test Plan Activities	3-1
3.1.4 System Test Plan Output.....	3-2
3.1.5 System Test Plan Exit Criteria.....	3-2
3.2 System Test Description Creation	3-2
3.2.1 System Test Description Entrance Criteria.....	3-2
3.2.2 System Test Description Input.....	3-2
3.2.3 System Test Description Activities.....	3-2
3.2.4 System Test Description Output	3-2
3.2.5 System Test Description Exit Criteria	3-2
4. CONFIGURATION MANAGEMENT FOR TEST EXECUTION	4-1
4.1 Software Unit Test	4-1
4.1.1 Software Unit Test Entrance Criteria.....	4-1
4.1.2 Software Unit Test Input.....	4-1
4.1.3 Software Unit Test Activities.....	4-1
4.1.4 Software Unit Test Output	4-1
4.1.5 Software Unit Test Exit Criteria	4-2
4.2 Integration Test	4-2
4.2.1 Integration Test Entrance Criteria.....	4-2
4.2.2 Integration Test Input.....	4-2
4.2.3 Integration Test Activities.....	4-2
4.2.4 Integration Test Output	4-3
4.2.5 Integration Test Exit Criteria	4-3

TABLE OF CONTENTS (Cont'd)

Section	Page
4.3 Performance Test	4-3
4.3.1 Performance Test Entrance Criteria	4-3
4.3.2 Performance Test Input	4-3
4.3.3 Performance Test Activities	4-3
4.3.4 Performance Test Output	4-4
4.3.5 Performance Test Exit Criteria	4-4
4.4 System Qualification Test	4-4
4.4.1 System Qualification Test Entrance Criteria	4-4
4.4.2 System Qualification Input	4-4
4.4.3 System Qualification Test Activities	4-4
4.4.4 System Qualification Test Output	4-4
4.4.5 System Qualification Test Exit Criteria	4-4
4.5 System Test Report	4-5
5. QUALITY ASSURANCE	5-1
5.1 Quality Assurance Overview	5-1
5.1.1 Evaluate the Corrective Action Process	5-2
5.2 Evaluate Configuration Management Process	5-2
5.2.1 Quality Assurance Activities	5-3
6. QUALITY ASSURANCE TEST ANALYSIS AND PLANNING	6-1
6.1 System Test Plan Overview	6-1
6.2 System Test Plan Creation	6-1
6.2.1 System Test Plan Entrance Criteria	6-1
6.2.2 System Test Plan Inputs	6-1
6.2.3 System Test Plan Activities	6-1
6.2.4 System Test Plan Outputs	6-2
6.2.5 System Test Plan Exit Criteria	6-2
6.3 Software Test Description Development	6-2
6.4 System Test Description Creation	6-2
6.4.1 Software Test Description Entrance Criteria	6-2
6.4.2 Software Test Description Inputs	6-2
6.4.3 Software Test Description Activities	6-3
6.4.4 Software Test Description Outputs	6-3
6.4.5 Software Test Description Exit Criteria	6-3
7. QUALITY ASSURANCE FOR TEST EXECUTION	7-1
7.1 Software Unit Test	7-1
7.1.1 Software Unit Test Entrance Criteria	7-1
7.1.2 Software Unit Test Input	7-1
7.1.3 Software Unit Test Activities	7-1
7.1.4 Software Unit Test Outputs	7-1
7.1.5 Software Unit Test Exit Criteria	7-1

TABLE OF CONTENTS (Cont'd)

Section	Page
7.2 Integration Test	7-2
7.2.1 Integration Test Entrance Criteria.....	7-2
7.2.2 Integration Test Inputs	7-2
7.2.3 Integration Test Activities.....	7-2
7.2.4 Integration Test Outputs	7-2
7.2.5 Integration Test Exit Criteria	7-2
7.3 Performance Test	7-3
7.3.1 Performance Test Entrance Criteria.....	7-3
7.3.2 Performance Test Inputs	7-3
7.3.3 Performance Test Activities.....	7-3
7.3.4 Performance Test Outputs.....	7-3
7.3.5 Performance Test Exit Criteria	7-3
7.4 System Qualification Test.....	7-3
7.4.1 System Qualification Test Entrance Criteria	7-4
7.4.2 System Qualification Test Inputs.....	7-4
7.4.3 System Qualification Test Activities	7-4
7.4.4 System Qualification Test Outputs	7-4
7.4.5 System Qualification Test Exit Criteria.....	7-4
GLOSSARY	Gls-1
BIBLIOGRAPHY	Bbl-1
APPENDIX A CONFIGURATION MANAGEMENT CHECKLIST	A-1
APPENDIX B SYSTEM PROBLEM REPORT TRACKING	B-1
APPENDIX C SYSTEM PROBLEM STATUS REPORT GUIDELINES	C-1
APPENDIX D TEST READINESS REVIEW REPORT GUIDELINES	D-1
APPENDIX E QUALITY ASSURANCE CHECKLISTS	E-1

Table of Figures

Figure	Page
Figure 2.1 SI&T Process Phases.....	2-1
Figure 5.1 SI&T Process Phases.....	5-1
Figure B.1 SPR Life Cycle Flow	B-3
Figure D.1 TRR Items.....	D-4

This page intentionally left blank

LIST OF ACRONYMS

CM	Configuration Management
CMO	Configuration Management Office
COTS	Commercial Off the Shelf
DCR	Document Change Request
CSCI	Computer Software Configuration Item
HWCI	Hardware Configuration Item
IRS	Interface Requirements Specification
QA	Quality Assurance
QAO	Quality Assurance Office
SD	System Development
SDF	Software Development File
SDP	System Development Plan
SFA	Office of Student Financial Assistance
SI&T	System Integration and Testing
SPR	System Problem Report
SQT	System Qualification Test
SRD	Software Requirements Document
SSS	System or Subsystem Specifications
STD	System Test Description
STO	System Test Organization
STP	System Test Plan
STR	System Test Report
SU	Software Unit
TRR	Test Readiness Review

This page intentionally left blank

1. INTRODUCTION

1.1 Background

The U.S. Department of Education, Office of Student Financial Assistance (SFA), contracted CTG, Incorporated (CTGi) in August, 2000 to develop standardized System Integration and Test (SI&T) procedures to be used for guidance, planning, and implementation of current and future Department of Education enterprise information technology systems projects.

1.2 Objective

The processes described in this document provide information and guidance to personnel concerned with Configuration Management (CM) and Quality Assurance (QA) during testing and integration phases of a system project. The intent of this document is to provide guidance to the practitioner so that each organization may adapt the processes, as appropriate, to create comprehensive CM and QA standards and procedures.

1.3 Applicability

When the SI&T process is performed by either the U. S. Department of Education, SFA, staff and/or contractors this document will apply, unless specifically excluded, in the program/project plan, contract, etc. This document will be used for the creation of guidelines and procedures for the planning, preparation, execution, analysis, and evaluation, of all types of U. S. Department of Education, SFA, information technology project integration and testing.

1.4 Assumptions

The procedures in this document represent the minimum of CM support that should be provided to the SI&T processes in lieu of policies established by the Configuration Management Office (CMO) and Quality Assurance Office (QAO).

The following assumptions are made:

- SFA has established standards and procedures for CM and QA, as defined in the Modernization Technology Handbook, and the CM and QA functions encompass the complete system life cycle of SFA individual development efforts.

- A CMO exists within the SFA. The CMO is responsible for providing CM support to the project manager for the SI&T process. The standards and procedures described within the CM section of this document and implemented by CM conform to the overall CMO schema.
- A QAO exists within the SFA. The QAO is responsible for providing QA support to the project manager for the SI&T process. The person(s) responsible for QA for a given system development effort will be assigned on a real-time basis.

1.5 Document Organization

This document is written to supply procedures and templates for two functions, CM and QA, performed during the SI&T process,. Basic guidelines, templates, and procedures to provide guidance in the areas of CM and QA are provided. Templates, checklists, and report details required implement the CM and QA functions are provided as appendices.

This document contains seven narrative sections, a Glossary, a Bibliography, and four appendices. Section 1, Introduction, provides brief background information and states the guiding objective and applicability for the document. Section 2, Configuration Management, provides an overview of the CM procedures to be followed during the testing life cycle. Section 3, Configuration Management For Test Analysis and Planning, provides procedures for conducting CM on the Test Analysis and Planning Phase of a project. Section 4, Configuration Management For Test Execution, provides procedures for providing CM for the test execution phases of the SI&T process. Section 5, Quality Assurance, provides an overview of QA procedures followed during the testing life cycle. Section 6, Quality Assurance For Test Analysis and Planning, provides procedures for conducting QA during the Test Analysis and Planning Phase of the SI&T process. Section 7, Quality Assurance For Test Execution, provides procedures for conducting QA during the test execution phases of SI&T process. Appendix A provides the Configuration Management Checklist. Appendix B provides a list of the types of data required to perform System Problem Report (SPR) tracking. Appendix C provides SPR guidelines. Appendix D provides information for the Test Readiness Review Report Guidelines. Appendix E provides QA Checklists.

2. CONFIGURATION MANAGEMENT

CM is a discipline that applies technical and administrative direction and control over the system life cycle. CM controls, identifies, and documents the functional and physical characteristics of a system. Testing is a subset of a systems life cycle. For the purpose of this section, the role of CM is limited to the SI&T process. The SI&T process defines the testing life cycle presented in Figure 2.1.

Baseline SRD	Phase 1	Phase 2	Phase 3	Phase 4	Phase 5	Customer Acceptance
	Baseline STP	SU Test	Integration Test	Performance Test	SQT Test	
	Baseline STD	Integration TRR	Performance TRR	SQT TRR	Final STR	

Figure 2.1 SI&T Process Phases

The CMO is the officiator of the CM system life cycle process. The CMO develops standards and procedures used to implement CM functionality throughout the test life cycle of the system. The procedures detailed in this section represent a minimum CM capability to support the SI&T processes in lieu of policies established by the CMO.

2.1 Configuration Management Overview

CM participation in the test life cycle consists of the following key elements:

- Baseline control.
- Version control and identification.
- Change control.
- Participation in review processes.

The CMO creates the procedures and standards used to execute the key elements listed above. Throughout the test life cycle, CM performs the following tasks to effectively administer the CMO procedures and standards:

- Ensure all formal software, hardware, and testing environment configurations specified in the Software Requirements Document (SRD) and System Test Plan (STP) are placed under formal CM control.
- Ensure all documentation is placed under CM control.
- Ensure the establishment of formal release procedures of CM approved software, documentation, and testing environment versions.
- Ensure the incorporation of all approved changes, to prevent unauthorized changes to the controlled Computer Software Configuration Item (CSCI), Hardware Configuration Item (HWCI), documentation, and testing environment.
- Report the status of system problems.

To assist in the CM process during the SI&T process, a CM checklist is provided in Appendix A. The items on this checklist are explained in further detail throughout the CM section of this document.

2.1.1 Participants

Participants needed for these tasks will vary based on the size and/or complexity of the system being tested. Participants will include government personnel designated/assigned by the SFA and/or the CMO. Participants are responsible for overseeing testing activities and providing documentation and answers to the SI&T team (contractor and/or government).

2.1.2 Baseline Control

Baseline control is the process that regulates approved versions of all CSCI, HWCI, documentation, and testing environments throughout the system life cycle. A baseline is a specification or product that is formally reviewed and agreed upon and, thereafter, serves as the basis for further development. A baseline can be changed “only” through formal change control procedures.

CM is responsible for ensuring the integrity of the baseline CSCI, HWCI, documentation, and testing environment. Traceability is provided throughout the testing life cycle by means of controls placed on the baseline system. CM ensures that all segments of the system are current, valid, and can be retrieved.

CM uses a baseline process to provide the administrative mechanism for establishing, initiating, preparing, evaluating, and approving all changes throughout the testing life cycle. The CMO develops standards and procedures for the baseline process. To implement the baseline control, CM executes the established standards and procedures manually or by automated processes.

CMO-established standards and procedures for baseline control should, at a minimum, include the following items:

- Control approved data.

- Control and monitor baseline content and quality.
- Develop a project library to store baseline and subsequent versions of software, documentation, and testing environments.
- Control the recovery and retrieval of baseline CSCI, HWCI information, documentation, and testing environment.

2.1.3 Version Control and Identification

The CMO develops standards and procedures for version control of CSCI, HWCI, documentation, and testing environments. After the CSCI, HWCI, documentation, and testing environments are baselined and placed under CM control, version control is established. When approved modifications are incorporated, a new version is created.

Throughout the testing life cycle, CM maintains version control manually or using automated tools. A function of version control is the process by which systems are tracked when changes are made to the code and it is recompiled and/or when the system environment is changed. Formal version control of CSCI, HWCI, documentation, and testing environments are usually generated at planned intervals in coordination with the testing schedule or system release schedule.

The CMO-established standards and procedures for version control should, at a minimum, include the following items:

- Compile approved baseline components.
- Control and monitor release content and quality.
- Report version implementation issues.
- Establish version identification.

A vital part of version control is version identification. Version identification is the means by which changes are identified and managed, while maintaining the integrity of the baseline. CM uses version identification to establish a unique identifier for all baseline CSCI, HWCI, documentation, and testing environments. The CMO is responsible for developing the standards and procedures for version identification. The CMO establishes and maintains version identification throughout the testing life cycle manually or using automated tools.

When a new version of a system is created, CM releases a version control report. The version control report informs the System Test Organization (STO) of additions, modifications and completed System Problem Reports (SPR) incorporated in the current version. In addition, a version control report cross-references and tracks testing against system requirements. The CMO is responsible for developing standards and procedures for the version control report. To

produce version control reports throughout the testing life cycle, CM executes established manually or uses automated tools. At a minimum, the following information is contained in the version control report:

- Testing life cycle phase.
- Date of report.
- Version identifier.
- List of fixed SPRs contained in the new version.
- List of changed CSCIs and HWCIs.
- Description of changes and enhancements to the CSCI, HWCI, documentation, and testing environments.
- Cross-reference of planned test execution against system requirements.
- Any other pertinent information regarding system and documentation updates.

2.1.4 Change Control

Change control is the process by which problems and changes to the CSCI, HWCI, documentation, and testing environment are evaluated, approved, scheduled, and tracked. During the testing life cycle, the STO uses SPRs to identify problems. Changes are also driven by modifications to software requirements. Based on CMO standards and procedures, the change control process is tracked manually or using automated tools. CM executes processes to implement change control throughout the testing life cycle.

CM creates a SPR status report and presents it at the meeting of the SPR Review panel. By participating in the SPR Review, CM reports the problems and changes made to the CSCI, HWCI, documentation, and testing environment to the STO. A recommended SPR Tracking process is presented in Appendix B. The SPR report is used to assist in determining if SPRs are being processed appropriately and expeditiously. CM continues to monitor and report on SPR status to all STO groups throughout the testing life cycle. The SPR status report can be produced manually or using automated processes. The SPR status report is presented in Appendix C and, at a minimum, contains the following information:

- Testing life cycle phase.
- Date of report.
- Date of last SPR Review.
- Reported by.

- Version identifier.
- Total number of SPRs submitted, assigned, opened, resolved, closed, postponed, duplicated, and re-opened itemized by severity.
- Grand total of SPRs to date.

2.1.5 Review Participation

SPR and test reviews are conducted throughout the testing life cycle. CM reports on the status of SPRs throughout the Integration Test, Performance Test, and System Qualification Test (SQT) phases. The system test groups (i.e., System Development (SD), Test, and QA) determine when completed SPRs are incorporated into the testing environment for re-testing.

Test reviews are conducted after the completion of each test phase to ensure communications among all system test groups. The Test Readiness Review (TRR) is used to determine if testing is complete and the system is ready for the next test phase. A detailed TRR report is shown in Appendix D.

This page intentionally left blank

3. CONFIGURATION MANAGEMENT FOR TEST ANALYSIS AND PLANNING

After system requirements are analyzed and documented in the SRD, or an equivalent document, Test Analysis and Planning begins. In the SRD, system requirements are itemized and uniquely identified to provide details needed to satisfy system requirements. CM baselines, assigns a version identifier, and places the SRD, or equivalent document, under version control.

From the SRD, test engineers create a STP. The STP describes the detailed approach for the Software Unit (SU), Integration, Performance, and SQT testing. CM baselines, assigns a unique identifier, and places the STP under version control.

After the STP is baselined, test engineers create the System Test Description (STD). The STD is used to implement the test strategy as defined in the STP. The STD describes the test preparations, test cases, and test case procedures necessary to perform all aspects of testing. CM baselines, assigns a version identifier, and places the STD under CM control.

It is important that both the STP and STD be baselined and under CM so that any revisions to the documents are managed and reported to the STO groups.

3.1 System Test Plan Creation

The following activities take place during the STP creation portion of the Test Analysis and Planning phase of the SI&T process and are monitored by the CM group.

3.1.1 System Test Plan Entrance Criteria

Reviewed and baseline SRD.

3.1.2 System Test Plan Input

- a. SRD approval process is complete.
- b. Baseline SRD.

3.1.3 System Test Plan Activities

- a. Create baseline for the STP.
- b. Place STP under version control.
- c. Assign version identification to the STP.
- d. Implement change control procedures.

3.1.4 System Test Plan Output

- a. Baseline STP with version identifier.
- b. Provide change status reports and notification of changes to system test groups.

3.1.5 System Test Plan Exit Criteria

- a. STP is complete.
- b. CMO procedures and standards are satisfied.

3.2 System Test Description Creation

The following activities will take place during this portion of the Test Analysis and Planning phase of the SI&T process and are monitored by the CM group.

3.2.1 System Test Description Entrance Criteria

Reviewed and baseline STP.

3.2.2 System Test Description Input

STP approval process is completed.

3.2.3 System Test Description Activities

- a. Create baseline for the STD.
- b. Place STD under version control.
- c. Assign version identification to the STD.
- d. Implement change control procedures.

3.2.4 System Test Description Output

- a. Baseline STD with version identifier.
- b. Provide change status reports and notification of changes to system test groups.

3.2.5 System Test Description Exit Criteria

- a. STD is complete and ready for test execution.
- b. CMO procedures and standards are met.

4. CONFIGURATION MANAGEMENT FOR TEST EXECUTION

CM participation in test execution begins with the creation of testing environments. Because the hardware and software requirements are in the SRD, CM can begin to build testing environments prior to completion of the SU Test, and the testing environments are ready when the SU Test phase is complete. The completed testing environment is baselined, a version identifier is assigned, and it is placed under version control.

CM is performed in the SU Test, Integration Test, Performance Test, and SQT phases. The following sections describe the CM participation in the four phases of testing.

4.1 Software Unit Test

The System Development (SD) group creates and tests the SUs and produces Software Development Files (SDF).

After the software is unit tested and approved, CSCI items are created, integrated, and tested. CSCIs are configured and a baseline is created. When the CSCI qualification testing is complete, the CSCIs and SDFs are placed under CM control. CM assigns a version identifier to the baseline configuration, places the qualification tested CSCI under version control, and sets up an approved integration or performance testing environment.

After the testing environment is assembled, CM participates in the Integration TRR.

4.1.1 Software Unit Test Entrance Criteria

None.

4.1.2 Software Unit Test Input

- a. Integrated CSCIs placed under version control and assigned a version identifier.
- b. Completed SDFs placed under version control.

4.1.3 Software Unit Test Activities

- a. Compile the integrated CSCIs.
- b. Baseline compiled integrated CSCIs.
- c. Baseline SDFs.
- d. Assign a version identifier to the baseline and place under version control.
- e. Participate in the Integration TRR.

4.1.4 Software Unit Test Output

Current version of CSCI prepared for Integration Test Phase.

4.1.5 Software Unit Test Exit Criteria

- a. SU testing is complete.
- b. CMO standards and procedures are met.
- c. Integration TRR is complete.

4.2 Integration Test

CM places the Integration Test phase-prepared software into the integration testing environment. The Integration Test phase is initiated and the STD test case(s) are executed. Problems are identified throughout the Integration Test phase and SPRs are completed and placed under CM control for tracking and reporting.

As problems are solved, new versions of the software are created, identified, and placed into the integration testing environment for re-testing, and the SPR status is updated accordingly. Prior to a new release of software, documentation or an Integration Testing environment, CM creates an SPR status report and participates in an SPR Review.

When the release is placed into the Integration Testing environment, CM provides a version control report. CM participates in test readiness reviews when integration testing is complete.

4.2.1 Integration Test Entrance Criteria

- a. Completion of the STD.
- b. Creation of testing environment.
- c. SUs integrated into CSCIs and tested by SD group.
- d. Procedures established for SPR tracking and reporting.
- e. Baseline CSCIs.
- f. Established version of the software.

4.2.2 Integration Test Input

The software version to be placed into the Integration Testing environment.

4.2.3 Integration Test Activities

- a. Baseline Computer Software Configuration Items (CSCIs) and the baseline integrity.
- b. Create and maintain Integration Testing environment.
- c. Transfer updated software releases to Integration Testing environment in accordance with version control and identification processes.
- d. Track and report status of SPRs in accordance with CM change control processes.
- e. Maintain and control baseline documentation.
- f. Produce version control reports as new versions are placed in the testing environment.
- g. Participate in SPR Review.
- h. Participate in Performance Test phase TRR.

4.2.4 Integration Test Output

- a. Current release of Performance Test phase approved software.
- b. Version control report.
- c. SPR status report.
- d. Approved System Test Report (STR).

4.2.5 Integration Test Exit Criteria

- a. Integration Test phase is complete.
- b. CMO standards and procedures are met.
- c. Performance Test phase TRR is complete.

4.3 Performance Test

CM places the Performance Test phase ready software into the performance testing environment. Performance Test phase is initiated and the STD test case(s) are executed. Problems are identified throughout the Performance Test phase and SPRs are complete and placed under CM control for tracking and reporting.

As problems are fixed, new versions of the software are created, identified, and placed into the performance testing environment for re-testing and the SPR status is updated accordingly. CM creates a SPR status report and participates in an SPR Review prior to release of a new version of software, documentation or performance testing environment. When a new version release is placed into the performance testing environment, CM provides a version control report. When performance testing is complete, CM participates in the TRR.

4.3.1 Performance Test Entrance Criteria

Creation of the Performance Test phase environment.

4.3.2 Performance Test Input

Software version to be placed into Performance Test phase environment.

4.3.3 Performance Test Activities

- a. Create and maintain Performance Test phase environment.
- b. Transfer updated software releases to Performance Test phase environment in accordance with version control and identification processes.
- c. Track and report status of SPRs in accordance with CMO change control processes.
- d. Maintain current version of documentation.
- e. Produce version control reports, as new versions are ready to be placed in the testing environment.
- f. Participate in SPR Review.
- g. Participate in the SQT phase TRR.

4.3.4 Performance Test Output

- a. Current version of SQT phase-approved software.
- b. Version control report.
- c. SPR status report.
- d. Approved STR.

4.3.5 Performance Test Exit Criteria

- a. Performance Test phase is complete.
- b. CMO standards and procedures are met.
- c. SQT phase TRR is complete.

4.4 System Qualification Test

SQT is the final phase of the SI&T process. It is conducted to ensure the final software version meets all functional and performance requirements. During this phase of testing, CM participation is limited to creating the SQT environment, providing a final production-ready version of the software, and forwarding production-ready reports to the system project manager and QA.

4.4.1 System Qualification Test Entrance Criteria

Creation of the SQT phase environment.

4.4.2 System Qualification Input

Software version to be placed into SQT phase environment.

4.4.3 System Qualification Test Activities

- a. Create and maintain SQT phase testing environment.
- b. Track and report status of SPRs in accordance with CMO change control processes.

4.4.4 System Qualification Test Output

- a. Release-ready version of software and documentation.
- b. Release-ready environment.
- c. Final STR.

4.4.5 System Qualification Test Exit Criteria

- a. SQT phase is complete.
- b. CMO standards and procedures are met.

4.5 System Test Report

CM participates in completing the system test report upon the completion of testing and the creation of baseline release-ready software, documentation, and environment.

This page intentionally left blank

5. QUALITY ASSURANCE

The purpose of QA is to provide adequate confidence that the system optimally fulfills customer expectations. This document will focus solely on the functions and actions of QA as they relate to the SI&T process as presented in Figure 5.1. The person(s) responsible for QA are referred to as the QA personnel.

	Phase 1	Phase 2	Phase 3	Phase 4	Phase 5	Customer Acceptance
	Baseline STP Baseline STD	SU Test Integration TRR	Integration Test Performance TRR	Performance Test SQT TRR	SQT Test Final STR	

Figure 5.1 SI&T Process Phases

5.1 Quality Assurance Overview

QA reviews and audits software test activities to verify compliance with applicable test procedures and standards and provides the project manager with the results. Evaluation of the test process is necessary to ensure the integrity of each test phase. Effective testing is the surest indication of software quality.

QA personnel participate in reviews and conduct audits during SI&T process phases to provide the STO with assurance that all test standards are being met. QA and software development personnel, at a minimum, perform reviews and/or audits to verify baseline stability and compliance with the test standards and procedures, during SI&T test phases.

The number of participants needed for these tasks will vary depending on the size and/or complexity of the system being tested and is determined by the project manager. To assist in the QA process, a QA checklist is provided in Appendix E.

The following sections explain the items on the QA checklist in further detail.

5.1.1 Evaluate the Corrective Action Process

QA invokes corrective actions to remedy problems found in the SI&T process. Problems include documentation errors, software errors, and noncompliance with standards and procedures. The SI&T organization uses SPRs to evaluate errors.

QA performs the following tasks:

- Periodic reviews of the corrective action processes and the results compared to SI&T standards.
- Periodic analysis of SPRs to identify trends that may disclose generic problem areas. These analyses include the study of causes, magnitude of impact, frequency of occurrence, and preventive measures.

QA performs the following corrective action processes:

- Identification and correction of problems that occur during software development for early detection of actual or potential problems.
- Report the problem(s).
- Analyze the problem(s) and propose corrective measures.
- Timely and complete corrective action.
- Record and follow-up status of each problem.

5.2 Evaluate Configuration Management Process

CM participation in the testing life cycle of a system consists of the following key elements:

- Baseline control.
- Version control and identification.
- Change control.
- Participation in review processes.

The CMO is responsible for creating the procedures and standards used to execute the key elements listed above. QA performs the following tasks throughout the testing life cycle to effectively administer CMO procedures and standards:

- Ensure all configurations, as specified in the SRD and STP, are placed under CM control. This includes software, hardware, testing environments, and documentation.
- Ensure formal release procedures for CM-approved software, documentation, and testing environment versions are established.

- Ensure incorporation of all approved changes to prevent unauthorized changes to controlled software, documentation, and testing environment.
- Report the status of system problems.

5.2.1 Quality Assurance Activities

QA is used to ensure access control of software is properly exercised and unauthorized changes to master files do not occur.

The following QA activities are performed during the CM process evaluation:

- Verify established standards for titling, naming, and describing change status are used for the configuration identification of documents, code, testing environment, and computer data.
- Verify that changes made to developmental baseline documents, code, testing environment, and computer data are identified, reviewed, implemented, and incorporated in accordance with established QA procedures.
- Verify configuration control of changes to baseline documents, software, and the testing environment is being managed in accordance with QAO requirements.
- Verify configuration status accounting reports are prepared at scheduled times and in accordance with established procedures.
- Verify personnel assigned to participate in configuration audits comply with QA requirements.
- Verify that the identification of all software includes the software name and a unique version identifier.

This page intentionally left blank

6. QUALITY ASSURANCE TEST ANALYSIS AND PLANNING

Test planning begins with the receipt of a system SRD. The SRD is used to draft an STP and an STD. During test planning phase, QA ensures the baseline STD and STP fulfill the criteria, as defined in the QA checklist for the SRD.

6.1 System Test Plan Overview

The STP describes the plans for testing a CSCI, a software system or subsystem, or both. There is usually a single STP for a system or project that describes the software testing environment, identifies tests to be performed, and provides a schedule for test activities. To develop a strategy that will produce the highest assurance of meeting customer requirements, the STP includes activities for SU Test, Integration Test, Performance Test, and SQT phases.

6.2 System Test Plan Creation

The following activities will take place during this portion of the Test Analysis and Planning phase of the SI&T process and are monitored by the QA group.

6.2.1 System Test Plan Entrance Criteria

System requirements are analyzed and the requirements for software development are documented in a baseline SRD.

6.2.2 System Test Plan Inputs

Baseline SRD.

6.2.3 System Test Plan Activities

- a. Verify software requirements are documented in the SRD.
- b. Verify test case(s), test classes (e.g., path analysis, stress, capacity, and timing), and test methodologies (e.g., static, dynamic) are defined.
- c. Verify testing environment(s) is identified.
- d. Verify plans for implementation and control of the testing environment are defined.
- e. Verify the number of personnel and other resources required, for implementing the testing process and objectives is defined.
- f. Verify a project test leader and the support personnel necessary to prepare, conduct, analyze, and report test results are assigned.
- g. Verify a test schedule is defined and includes time for problem correction and re-testing.

- h. Verify creation of the STP is in accordance with the instructions of QAO documentation standards.
- i. Participate in draft STP peer review.
- j. Verify draft STP is revised to correct discrepancies and incorporates approved changes.
- k. Verify STP is approved and under CM control.

6.2.4 System Test Plan Outputs

- a. QA reviews for completeness.
- b. QA checklist is complete.

6.2.5 System Test Plan Exit Criteria

Forward status report, with QA checklist attached, to CM and project manager.

6.3 Software Test Description Development

The STD describes test preparations, test cases, and test case procedures to test a software system, subsystem, or both. The STD enables the customer to assess the adequacy of the testing to be performed.

Test case procedures are defined as a series of individually numbered steps listed in the order the steps will be performed. The appropriate level of detail in each test case procedure depends on the type of software being tested. For some software, each keystroke may be a separate step in the test case procedure; however, for most software each step includes a series of logically related keystrokes or actions.

6.4 System Test Description Creation

The following activities will take place during this portion of the Test Analysis and Planning phase of the SI&T process and are monitored by the QA group.

6.4.1 Software Test Description Entrance Criteria

System requirements are analyzed and requirements allocated for software development are documented in a baseline SRD.

6.4.2 Software Test Description Inputs

- a. Baseline SRD.
- b. STP.

6.4.3 Software Test Description Activities

- a. Verify the SRD has corrected discrepancies and approved changes.
- b. Verify the required individual test cases are identified for each test phase.
- c. Verify test cases procedures are written to support each test case.
- d. Verify the expected result(s) (e.g., output from test in response to input) that will serve as the test case for the Pass/Fail criteria are defined for each test case.
- e. Verify test case procedures, including inputs and expected results, are documented.
- f. Verify inputs (stimuli) required to fulfill the test purpose are defined for each test case.
- g. Verify data insertion and extraction methods to/from test is defined.
- h. Verify the points of data input/output and amount of data needed are identified.
- i. Verify evaluation criteria for test result(s) analysis is defined to include ranges of values, capacities, and times for testing Pass/Fail status.
- j. Verify SRD performance requirement(s) to a specific test case(s) is traced.
- k. Verify identification of testing environment configuration, interface drivers, database loaders, controllers/monitors, other test tools, etc., for support of test case(s).
- l. Verify draft STD is prepared for review.
- m. Participate in peer review.
- n. Verify test case procedure data file(s) is compiled.
- o. Verify STD is approved and under CM control.

6.4.4 Software Test Description Outputs

- a. Baseline audit of STD.
- b. QA reviews for completeness.
- c. QA checklist is complete.

6.4.5 Software Test Description Exit Criteria

Forward status report, with QA checklist attached, to CM and project manager.

This page intentionally left blank

7. QUALITY ASSURANCE FOR TEST EXECUTION

The Test execution stage encompasses the process whereby actual, hands-on testing is performed, whether it is accomplished manually or using automated test tools.

The four phases of test execution are SU Test, Integration Test, Performance Test, and SQT. The following sections describe QA participation in each of the four phases of testing.

7.1 Software Unit Test

During the SU Test phase of the SI&T process, QA ensures that the SDF is updated and CSCI qualification testing is executed in accordance with testing procedures and guidelines.

7.1.1 Software Unit Test Entrance Criteria

Completion of the unit and unit integration portions of the SU Test phase.

7.1.2 Software Unit Test Input

Baseline SRD.

7.1.3 Software Unit Test Activities

- a. Verify version-specific test results are documented in the SDF.
- b. Verify test logs are complete.
- c. Participate in integration TRR.

7.1.4 Software Unit Test Outputs

- a. Baseline audit.
- b. QA reviews for completeness.
- c. QA checklist is complete.
- d. Baseline SDF

7.1.5 Software Unit Test Exit Criteria

- a. Forward status report, with QA checklist attached, to CM and project manager.
- b. Tested CSCI submitted to CM.
- c. TRR approval to proceed to Integration Test phase.

7.2 Integration Test

Integration testing is the test phase where the focus shifts from individual component correctness to the proper operation of interfaces between components, the flow of information through the system, and the satisfaction of system requirements. To complete software and system functionality testing, software integration and test activities combine individually developed CSCI modules in the testing environment to verify they function together. For joint hardware and software development, integration requires close synchronization of hardware and software to meet designated integration and testing milestones.

Part of the Integration Test phase is regression testing. Regression testing is performed after CSCI integration testing is complete to confirm modifications do not affect the functions of the original system.

7.2.1 Integration Test Entrance Criteria

- a. SU Test phase is complete.
- b. TRR approval to proceed to the Integration Test phase.

7.2.2 Integration Test Inputs

- a. Baseline STP.
- b. SDF, if needed.
- c. Baseline STD.
- d. Integration Testing environment confirmed.

7.2.3 Integration Test Activities

- a. Verify test case procedure(s) are conducted properly and test result(s) are recorded as they are observed.
- b. Verify required post-test analysis or data reduction to determine Pass/Fail criteria, as specified in the test case procedure, is performed.
- c. Verify the Integration Test phase STR is reviewed and approved.
- d. Verify SPRs are submitted for all detected problems.
- e. Participate in Performance Test phase TRR.

7.2.4 Integration Test Outputs

- a. Integration Test phase STR is complete.
- b. Baseline audit.
- c. QA reviews for completeness of test phase.
- d. QA checklist is complete.

7.2.5 Integration Test Exit Criteria

- a. Forward status report, with QA checklist attached, to CM and project manager.
- b. Integration Test phase STR is complete.
- c. Tested system under CM.
- d. TRR approval to proceed to Performance Test phase.

7.3 Performance Test

Performance testing validates the compatibility of all hardware and software components in accordance with the SRD. The Performance Test phase is conducted in an environment simulating real-world usage.

7.3.1 Performance Test Entrance Criteria

- a. Software and hardware under CM.
- b. TRR approval to proceed to the Performance Test Phase.

7.3.2 Performance Test Inputs

- a. Production configuration and environment confirmed.
- b. Baseline STP.
- c. Baseline STD.
- d. SPR from Integration Test phase, if needed.

7.3.3 Performance Test Activities

- a. Participate in TRR for SQT Test phase.
- b. Verify test case procedures are conducted properly and test results are recorded as they are observed.
- c. Verify required post-test analysis or data reduction to determine Pass/Fail criteria, as specified in the test case procedure, is performed.
- d. Verify Performance Test phase STR is reviewed and approved.
- e. Verify SPRs are submitted for all detected problems.

7.3.4 Performance Test Outputs

- a. Performance Test phase STR is complete.
- b. Baseline audit.
- c. QA reviews for completeness.
- d. QA checklist is complete.

7.3.5 Performance Test Exit Criteria

- a. Forward status report, with QA checklist attached, to CM and system project manager.
- b. Performance Test phase STR is complete.
- c. Tested system under CM.
- d. TRR approval to proceed to SQT Phase.

7.4 System Qualification Test

The SQT is conducted to ensure that the final software meets all customer/user functional and performance requirements.

7.4.1 System Qualification Test Entrance Criteria

- a. Software and hardware under CM.
- b. TRR approval to proceed to SQT Phase.

7.4.2 System Qualification Test Inputs

- a. Test case(s) and test case procedure(s).
- b. Production configuration and environment confirmed.

7.4.3 System Qualification Test Activities

- a. Verify test case procedures are conducted properly and test results are recorded as they are observed.
- b. Verify required post-test analysis or data reduction to determine Pass/Fail criteria, as specified in the test case procedure, is performed.
- c. Verify Final STR is reviewed and approved.
- d. Verify SPRs are submitted for all detected problems.

7.4.4 System Qualification Test Outputs

Final STR is complete.

7.4.5 System Qualification Test Exit Criteria

- a. Final STR.
- b. Completion of all outstanding SPRs.
- c. System acceptance by project manager or customer.

GLOSSARY

Aggregate

A mass of distinct things gathered into a total or whole.

Aggregation Level

Effective measurement analysis and reporting requires that the data be aggregated to higher levels of the of the software components and project organizational structure. The aggregation levels define the different ways the measurement data can be grouped and organized for reporting on the project. The aggregation levels describe how the measurement data relates to an existing product and process structures. The organization that allows the measurement results to be combined, and later decomposed, into meaningful pieces of information.

Aggregation Structure

The structure used to define the data according to the defined aggregation levels. The levels may describe the personnel and management structure of the project, or the configuration of physical components of the project. All entries in a structure should be of the same type, such as software modules. However, these entries may reside at various levels of the structure, such as software modules at the unit level, CSCI, or integrated level of the software architecture.

Application

(1.) A complete, self-contained program that performs specific function(s) directly for the user.

(2.) In the TPM process this term refers to one of the two basic measurement activities which comprise the system measurement process. The application activity involves collecting, analyzing, and acting upon the measurement data.

See **Tailoring**.

Automated Test Script

A computer readable set of instructions that performs a sequence of steps, sub-steps, or other actions, performed serially, in parallel, or in some combination of consecution, that creates the desired test conditions that the test case is deigned to evaluate.

Baseline

A specification or product that has been formally reviewed and agreed upon, that thereafter serves as the basis for further development, and that can be changed only through formal change control procedures.

Baseline Control

Baseline control is the process that regulates approved and released versions of all software, documentation, and the test environment throughout the test life cycle.

Black Box Testing

This is testing associated with functional testing where the object being tested is treated as a black box. In this type of testing the test object is subjected to inputs and outputs that are verified for conformance to prescribed specifications.

Capacity Testing

Attempts to simulate expected customer peak load operations in order to ensure that the system performance requirements are met. It does not necessarily exercise all of the functional areas of the system, but selects a subset that is easy to replicate in volume. It will ensure that functions which are expected to use the most system resources are adequately represented.

Change Control

The process by which problems and changes to the software, documentation, and test environment are evaluated, approved, rejected, scheduled, and tracked.

Computer Aided Software Engineering (CASE)

A technique for using computers to help with one or more phases of the software life cycle, including the systematic analysis, design, implementation and maintenance of software. Adopting the CASE approach to building and maintaining systems involves software tools and training for the developers who will use them.

Computer Software Configuration Item (CSCI)

An aggregation of software that is designated for configuration management and treated as a single entity in the configuration management process.

Configuration Control

An element of configuration management, consisting of the evaluation, coordination, approval or disapproval, and implementation of changes to configuration items after formal establishment of their configuration identification.

Configuration Item (CI)

Hardware or software, or an aggregate of both, which is designated by the project configuration manager (or contracting agency) for configuration management.

Configuration Management (CM)

A discipline applying technical and administrative direction and surveillance to: identify and document the functional and physical characteristics of a configuration item, control changes to those characteristics, record and report change processing and implementation status, and verify compliance with specified requirements.

Configuration Management Office (CMO)

The Configuration Management Office (CMO) is the officiator of the project life cycle CM process.

Criteria

A standard, rules, or tests by which something can be judged.

Critical Defect

See Criticality

Criticality

The assessment of the impact upon a system of a given error, defect, problem, or discrepancy during the life cycle of a system.

The definition of critical and non-critical system defects or problems should be addressed at a management level and can be different for each system. For any given system error, defect, problem, or discrepancy, an appropriate impact value (i.e., priority) will be assigned.

An example of impact values with the corresponding priority numbers is presented below as contained in IEEE/EIA Std-12207, 1998. The priority that will apply if a problem can result in one or more of these impacts:

PRIORITY	IMPACT
1.	a.) Prevent the accomplishment of an operational or mission essential capability. b.) Jeopardize safety. c.) Cause significant technical, cost, or schedule risks to the project or to life cycle support of the system.
2.	a.) Adversely affect the accomplishment of an operational or mission essential capability and no work-around solution is known. b.) Adversely affect technical, cost, or schedule risks to the project or to life cycle support of the system, and no work-around is known.
3.	a.) Adversely affect the accomplishment of an operational or mission essential capability, but a work-around solution is known. b.) Adversely affect technical, cost, or schedule risks to the project or to life cycle support of the system, but a work-around is known.
4.	a.) Results in user/operator inconvenience or annoyance, but does not affect a required operational or mission essential capability. b.) Results in inconvenience or annoyance for development or support personnel, but does not prevent the accomplishment of the responsibilities of these personnel.
5.	a.) This priority denotes any other effect.

Customer

The organization that procures software systems for itself or another organization.

Developer

An organization that develops software products. The term “develop” may include develop, modification, integration, reengineering, sustaining engineering, maintenance, or any other

activity that results in software products. The developer may be a contractor or a government agency.

Discrepancy

An inconsistency or disagreement found during testing between the actual and expected test results.

Document

A data medium and the data recorded on it that generally has permanence and can be read by a human operator or machine. Often used to describe human readable items only (e.g., technical documents, design documents, requirements documents, etc.).

Documentation

- (1.) A collection of documents on a given subject.
- (2.) The management of documents, that includes the actions of identifying, acquiring, processing, storing, and disseminating.
- (3.) Any written or pictorial information describing, defining, specifying, reporting or certifying activities, requirements, procedures, or results.

Driver

A software program that exercises a system or system component by simulating the activity of a higher level component.

Emulation

One system is said to emulate another when it performs in exactly the same way, though perhaps not at the same speed. A typical example would be the emulation of one computer by (a program running on) another. You might use emulation, as a replacement for a system whereas you would use a simulation if you just wanted to analyze it and make predications about it.

Emulator

Hardware or software that performs emulation.

Entry Criteria

A set of decision making guidelines used to determine whether a system under test is ready to move into, or enter, a particular phase of testing. Entry criteria tend to become more rigorous as the test phases progress.

Environment

The infrastructure in which a system is executing, consisting of hardware, operating system software, interfaces, etc.

Exit criteria

A set of decision-making guidelines used to determine whether a system under test is ready to exit a particular phase of testing. When exit criteria are met, either the system under test moves on to the next test phase or the test project is considered complete. Exit criteria tend to become more rigorous as the test phases progress.

Final System Test Report (FSTR)

Used to determine whether system testing is completed and to assure that software is ready for production.

Hardware Configuration Item (HWCI)

An aggregation of hardware that is designated for configuration management and treated as a single entity in the configuration management process.

Independent Verification and Validation (IV&V)

The verification and validation of a software product by an organization that is both technically and managerially separate from the organization responsible for developing the product.

Indicator

A measure or combination of measures that provides insight into a system issue or concept. TPM frequently uses indicators that are comparisons, such as planned versus actual measures. Indicators are generally presented as graphs or tables.

Integration

Combining software or hardware components or both into an overall system.

Integration Testing

The period of time in the software lifecycle during which the application is tested in a simulated production environment to validate the communications and technical architecture of the system. This test phase occurs when all the constituent components of the system under test are being integrated.

Interactive Development Environment (IDE)

A system for supporting the process of writing software. Such a system may include a syntax-directed editor, graphical tools for program entry, and integrated support for compiling and running the program and relating compilation errors back to the source code.

Interface

- (1.) A shared boundary (e.g., a hardware component linking two devices or registers, or a portion of storage accessed and/or modified by two or more computer programs).
- (2.) To interact or communicate with another system component.

Interface Requirement

A requirement that specifies a hardware, software, or database element with which a system or system component must interface, or that sets forth constraints caused by such an interface.

Interface Specification

A specification that sets forth the interface requirements for a system or system component (e.g., the software interface specification document).

Interface Testing

Tests conducted to ensure that program or system components correctly pass data and/or control to one another.

Issue

An area of concern where obstacles to achieving program objectives might arise. Issues include risks, problems, and lack of information. These three types of issues are defined as:

- Risk -- An area of concern that could occur, but is not certain. A risk is a potential problem. Risks represent the potential for the realization of unwanted, negative consequences from a project event. For example, a project plan may be based on the assumption that a COTS component will be available on a given date. There is a possibility (probability) that the COTS may be delayed and have some amount of negative impact on the project.
- Problem -- An area of concern that a project is currently experiencing or is relatively certain to experience. For example, a shortage of staff with the right skills may be an actual problem that is delaying the project.
- Lack of Information -- An area where the available information is inadequate to reliably predict project impact. Thus, satisfaction of project objectives is questionable even if no problems or risks are present. For example, lack of information about the size of the software to be developed could result in the project “discovering” that it has more work to do than originally planned.

Measure

The result of counting or otherwise quantifying characteristics of a process or product. Measures are numerical values assigned to system attributes according to defined criteria.

Measured (or actual) Value

Actual, current measurement data, such as hours of effort expended or line of code produced.

Measurement

The process of assigning quantitative values of system properties, according to some defined criteria. This process can be based on estimation or direct measurement. Estimation defines planned or expected measures. Direct measurement results in actual measures.

Measurement Analysis

The uses of measurement data to identify problems, assess problem impact, project an outcome, or evaluate alternatives related to system issues.

Measurement Analyst

The person(s) or team responsible for tailoring and applying system measures for a given project or task.

Measurement Information

Knowledge derived from analysis of measurement data and measurement indicators.

Milestone

A scheduled event for which some project or task member or manager is held accountable. A milestone is often used to measure progress.

Module

A program unit that is discrete and identifiable with respect to compiling, combining with other units, and loading.

Note: *The terms 'module', 'component', and 'unit' are often used interchangeably or defined to be sub-elements of one another in different ways depending on the context.*

Non-Critical Defect

See Criticality

Performance Testing

The period of time in the system or software development lifecycle during which the response times for the application are validated to be acceptable. The tests ensure that the system environment will support production volumes, both batch and on-line.

Priority

A measure of the elements of importance related to the repair of a system problem that are not considered in defining the severity of a system problem.

Project Manager (PM)

The official responsible for acquiring, developing, or supporting a system to meet technical, cost, schedule, and quality requirements. Acquisition, development, and support will include both internal tasks and work that is contracted to another source.

Quality Assurance (QA)

A planned and systematic pattern of all actions necessary to provide adequate confidence that the product optimally fulfils customers expectations.

Quality Control (QC)

The assessment of product compliance. Independently finding deficiencies assures compliance of the product with stated requirements.

Requirement

- (1.) A condition or capability needed to solve a problem or achieve an objective.
- (2.) A condition or capability that must be met or possessed by a system or system component to satisfy a contract, standard, specification, or other formally imposed document. The set of all requirements forms the basis of development.

Regression testing

Part of the test phase of software development where, as new modules are integrated into the system and the added functionality is tested, previously tested functionality is re-tested to assure that no new module has corrupted the system.

Risk

An area of concern that may occur, but is not certain. A risk is a potential problem. Risks represent the potential for the realization of unwanted, negative consequences from a project event. For example, a project plan may be based on the assumption that a commercial off the shelf (COTS) component will be available on a given date. There is a possibility (probability) that the COTS may be delayed and have some amount of negative impact on the project.

Severity

The degree to which a problem adversely influences the system's operation or the overall test effort.

Simulation

Attempting to predict aspects of the behavior of a system by creating an approximate (mathematical) model of it. This can be done by physical modeling, by writing a special-purpose

computer program or using a more general simulation package, aimed at a particular kind of simulation. Typical examples are aircraft simulators or electronic circuit simulators.

Simulator

Hardware or software that performs simulation.

Software Design Specification (SDS)

A document that records the design of a system or system component; typical contents include: system and/or component algorithms, control logic, data structures, data set use, input/output formats, and interface descriptions.

Software Development File (SDF)

The developer shall document the development of each Computer Software Unit (CSU), Computer Software Component (CSC), and CSCI in Software Development Files (SDF). The developer shall establish a separate SDF for each CSU or a logically related group of CSUs, for each CSC or a logically related group of CSCs, and for each CSCI. The developer shall document and implement procedures to establish and maintain SDFs. SDFs may be generated, maintained, and controlled by automated means. To reduce duplication, SDFs should not contain information provided in other documents or SDFs. The set of SDFs shall include (directly or by reference) the following information:

- Design considerations and constraints.
- Design documentation and data.
- Scheduling and status information.
- Test requirements and responsibilities.
- Test case, test case procedures, and results.

Software Life Cycle

The phases a software product goes through between when it is conceived and when it is no longer available for use. The software life cycle typically includes the following: requirements, analysis, design, construction, testing (validation), installation, operation, maintenance, and retirement. The development process tends to run iteratively through these phases rather than linearly; several models (spirals, waterfall, etc.) have been proposed to describe this process. Other processes associated with a software product are: quality assurance, marketing, sales, and support.

Software Management Plan

A project plan for the development of the software component of a system or for the development of a software product.

Software Requirements Document (SRD)

This is a formal document derived from the Software Requirements Specification (SRS) that sets forth the requirements, specifications, and standards for a system (e.g., a software product). Typically included are functional specifications and requirements, performance specifications and requirements, interface specifications and requirements, design specifications and requirements, and development requirements and standards.

Software Requirements Specification (SRS)

A specification that sets forth the requirements for a system component; (e.g., a software product). Typically included are functional requirements, performance requirements, interface requirements, design requirements, and development standards.

Software Tool

Computer programs used to help develop, test, analyze, or maintain another computer program or its documentation.

Specification

Documentation containing a precise, detailed, verifiable description of particulars with respect to the requirements, design, function, behavior, construction, or other characteristics of a system or system component.

Stub

- (1.) A dummy procedure used when linking a program with a run-time library. The stub routine need not contain any code and is only present to prevent “undefined label” errors at link time.
- (2.) A local procedure in a remote procedure call (RPC). The client calls the stub to perform some task and need not necessarily be aware that RPC is involved. The stub transmits parameters over the network to the server and returns the results to the client/caller.

System

- (1.) Any large program.
- (2.) The entire computer system, including the input/output devices, supervisor program or operating system and possibly other software.

System Problem Report (SPR)

A form that is used to record a discrepancy discovered during the Integration Test, Performance Test and System Qualification Test phases of the SI&T process concerning a Computer Software Configuration Item, a software system or subsystem, other software related items, and associated documentation.

System Problem Report (SPR) Status Report

The System Problem Report Status Report is used during the SPR Status Review to determine if the SPRs are being processed appropriately and expeditiously.

System Testing

The period of time in the software lifecycle during which the implementation of each requirement is validated.

Tailoring

In the TPM process, this term refers to one of the two basic measurement activities, which comprise the system measurement process. The tailoring activity includes identification and prioritization of program issues, selection and specification of appropriate system measures, and integration of the measurement requirements to the developer's system process.
See **Application**.

Test

The process of exercising a product to identify differences between expected and actual behavior.

Test Artifacts

An item created during the system integration and test process that is preserved upon completion of the test process (e.g., test plans, requirements documentation, automated test scripts, and test documentation).

Test Case

A description of a test to be executed for or focused on a specific test aim.

Test Case Procedures

A sequence of steps, sub-steps, and other actions, performed serially, in parallel, or in some combination of consecution, that creates the desired test conditions that the test case is designed to evaluate.

Test Case (Setup) Suite

The steps required to configure the test environment for execution of a test case.

Testing Condition

System state or circumstance created by proceeding through some combination of steps, sub-steps, or actions in a test case.

Testing Environment

The infrastructure in which the test is performed, consisting of hardware, system software, test tools, and procedures.

Test Plan

In a test plan the general structure and the strategic choices with respect to the test to be executed are formulated. The test plan forms the scope of reference during execution of the test and also serves as an instrument to communicate with the customer of the test. The test plan is a description of the test project, including a description of the activities and planning, therefore it is *not* a description of the tests themselves.

Test Readiness Review (TRR)

Review conducted to determine whether a software test phase has been completed and to assure that the software is prepared for the next step in the formal integration and testing procedures. Software test procedures and results are evaluated, for compliance with the software testing requirements and system descriptions, for adequacy in accomplishing testing goals. Also, provides the forum for updating and revising operational and supporting documentation.

Test Resources

Aids that are used by a test tool for collecting, tracking and controlling information. This information is:

- Software requirements defined in the Software Requirements Document.
- Test requirements defined in the System Test Description.
- Automated test case scripts as defined in the System Test Description.
- SPRs as determined at each phase of the System Integration and Testing process.

This information is controlled by Configuration Management at the end of the SI&T process for use whenever further testing may be conducted, using a testing tool, during the remaining lifecycle of the software or system.

Test Tools

The software, hardware, systems, or other instruments that are used to measure and test an item.

Traceability

Degree to which a relationship can be established between two or more products of the development process, especially products having a predecessor, successor, or master-subordinate relationship to one another (e.g., the degree to which the requirements and design of a given software component match).

Unit

The lowest element of a software hierarchy that contains one or more of the following characteristics:

- (1) A unit comprising one or more logical functional entities.
- (2) An element specified in the design of a computer software component that is separately testable.
- (3) The lowest level to which software requirements can be traced.

- (4) The design and coding of any unit can be accomplished by a single individual within the assigned schedule.

Unit Test

The process of ensuring that the unit executes as intended. This usually involves testing all statements and branch possibilities.

Version

One of a sequence of copies of a system, each incorporates new modifications.

Version Identifier

A unique identifier assigned to baseline software, documentation, and test environment.

Version Control

The process by which all changes to the software, documentation, and test environment are compiled and built into a new version of the system.

Version Control Report

A report that details all changes and enhancements made to current version of the software, documentation, and test environment.

White Box Testing

This type of testing is associated with structural testing in which the testing can be characterized as being tied to implementation details, such as control methods, database design, coding details, and logic paths. The process of how an individual input is treated to produce a given output is ascertained. Structural testing is sometimes referred to as “clear box testing” since white boxes are considered opaque and do not really permit visibility into the code.

Work Breakdown Structure (WBS)

A work breakdown structure for software defines the software-related elements associated with program work, work activities, and products. Many measures are aggregated and analyzed at various WBS levels.

BIBLIOGRAPHY

Black, Rex. *Managing The Testing Process*, Redman, WA: Microsoft Press, 1999

Koomen, Tim and Pol, Martin. *Test Process Improvement*, Essex, England, UK: Pearson Education Limited, 1999

Evans, Michael and Marciniak, John. *Software Quality Assurance and Management*, New York, NY: John Wiley & Sons, 1987

Carnegie Mellon University, Software Engineering Institute. *The Capability Maturity Model: Guidelines for Improving the Software Process*, 1995.

Institute of Electrical and Electronics Engineers (IEEE). "Glossary of Software Engineering Terminology", IEEE-Std-610.12, 1990.

Institute of Electrical and Electronics Engineers (IEEE)/Electronic Industries Alliance (EIA). "Software Life Cycle Processes", IEEE/EIA Std-12207, 1998.

U.S. Department of Education. "SFA System Integration & Testing Approach, SFA Modernization", Undated.

U.S. Department of Education. "SFA Enterprise Configuration Management Approach, SFA Modernization", Undated.

U.S. Department of Education, SFA. "Test Performance Measurements", December 2000.

U.S. Department of Education, SFA. "Procedures and Templates for Test Creation", November 2000.

U.S. Department of Education, SFA. "Procedures and Templates for Test Execution", November 2000.

U.S. Department of Education, SFA. "Procedures and Templates for Configuration Management and Quality Assurance", December 2000

U.S. Department of Education, SFA. "Procedures for Testing Tool", November 2000

Federal Systems Integration and Management Center (FEDSIM). "FEDSIM Writers Guide, Version 2", May 1994

Practical Software Measurement: A Foundation For Objective Project Management, Version 3.1,
17 April 1998, Office of the Under Secretary of Defense For Acquisition and Technology, Joint
Logistics Commanders Joint Group On Systems Engineering

Free On-Line Dictionary Of Computing web site at www.foldoc.org

APPENDIX A

CONFIGURATION MANAGEMENT CHECKLIST

This page intentionally left blank

Configuration Management Checklist

Step/Task Number	Description	Responsible Party or Parties	Completed Date
Part 1	CMO Standards and Procedures		
Step 1	CMO Baseline Standards and Procedures are complete.		
Step 2	CMO Version Control and Identification Standards and Procedures are complete.		
Step 3	CMO Change Control Standards and Procedures are complete.		
Part 2	Test Planning		
Step 1	STP		
Task 1	1. Create baseline for the STP.		
	2. Place STP under version control.		
	3. Assign version identification to the STP.		
	4. Implement change control procedures.		
	5. Provide change control status reports and notification of changes to system test groups.		
	6. Meet CMO Standards.		
Step 2	STD		
Task 2	1. Create baseline for the STD.		
	2. Place STD under version control.		
	3. Assign version identification to the STD.		
	4. Implement change control procedures.		
	5. Provide change control status reports and notification of changes to system test groups.		
	6. Meet CMO Standards.		
Part 3	Test Execution		
Step 1	Software Unit Test		
Task 1	1. Compile integrated CSCIs.		
	2. Baseline compiled integrated CSCIs.		
	3. Baseline SDFs.		
	4. Assign a unique identifier to the baseline and place under version control.		
	5. Participate in the integration test review and completion of the integration TRR.		
	6. CMO standards and procedures were met.		

Step/Task Number	Description	Responsible Party or Parties	Completed Date
Step 2	Integration Test		
Task 1	1. Baseline CSCIs and maintain baseline integrity.		
	2. Create and maintain integration testing environment.		
	3. Transfer updated software releases to integration testing environment in accordance with version control and identification processes.		
	4. Track and report status of SPRs in accordance with CM processes.		
	5. Maintain and control baseline documentation.		
	6. Produce version control reports as new versions are ready to be placed in the testing environment.		
	7. Participate in SPR Review.		
	8. Participate in performance TRR.		
	9. CMO standards and procedures were met.		
Step 3	Performance Test		
Task 1	1. Create and maintain performance testing environment.		
	2. Transfer updated software releases to performance testing environment in accordance with version control and identification processes.		
	3. Track and report status of SPRs in accordance with CMO processes.		
	4. Maintain current version of documentation.		
	5. Produce version control reports as new versions are ready to be placed in the testing environment.		
	6. Participate in SPR Review.		
	7. Participate in the SQT TRR.		
	8. CMO standards and procedures were met.		

Step/Task Number	Description	Responsible Party or Parties	Completed Date
Step 4	SQT Test		
Task 1	1. Create and maintain SQT testing environment.		
	2. Track and report status of SPRs in accordance with CMO processes.		
	3. Complete of the Final System Test Report.		
	4. CMO standards and procedures were met.		
Step 5	Final System Test Report Participation		

This page intentionally left blank

APPENDIX B

SYSTEM PROBLEM REPORT TRACKING

This page intentionally left blank

System Problem Report Tracking

System problems can be discovered during any phase of testing and are reported as they are discovered. Once a problem is discovered, a System Problem Report (SPR) form is completed. The SPR is then entered into a tracking database and controlled throughout the life of the project. The entire SPR tracking process is depicted in Figure B-1.

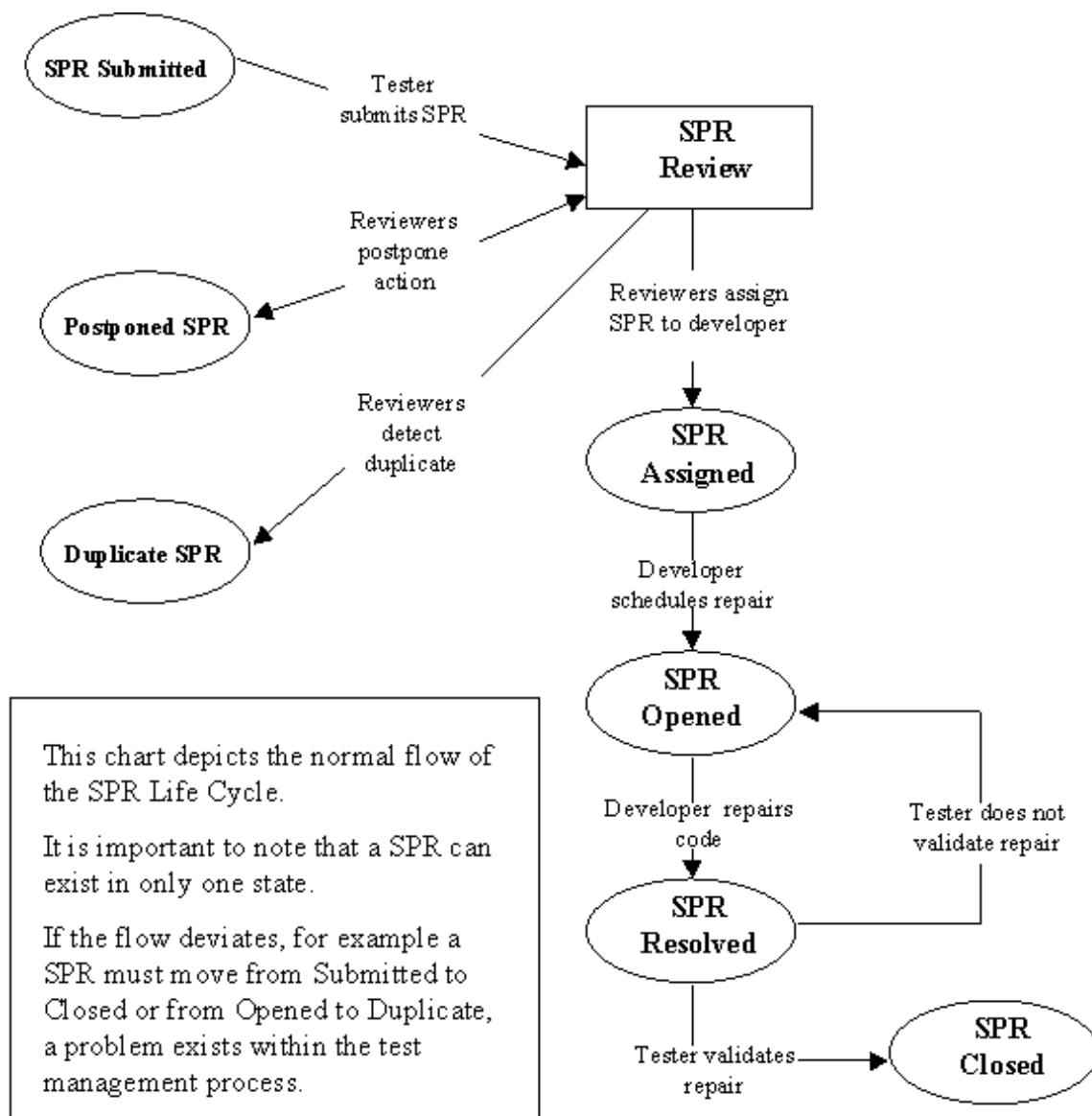


Figure B.1 SPR Life Cycle Flow

Submitted SPRs pass through a formal review process referred to as the SPR Review. Representatives from each group within the software test organization participate in the SPR Review. Representatives from the SD group and the Test group jointly determine the risk each SPR poses. An assessment is made to determine the severity of the problem and the priority for repairing the problem.

The SPR review group uses priorities to express the level of urgency for repair of the problem. The definition of critical and non-critical system defects or problems should be addressed at a management level and can be different for each system. For any given system error, defect, problem, or discrepancy, an appropriate impact value (i.e., priority) will be assigned.

An example of impact values with the corresponding priority numbers is presented below as contained in IEEE/EIA Std-12207, 1998.

The priority that will apply if a problem can result in one or more of these impacts:

PRIORITY

IMPACT

1.
 - a.) Prevent the accomplishment of an operational or mission essential capability.
 - b.) Jeopardize safety.
 - c.) Cause significant technical, cost, or schedule risks to the project or to life cycle support of the system.
2.
 - a.) Adversely affect the accomplishment of an operational or mission essential capability and no work-around solution is known.
 - b.) Adversely affect technical, cost, or schedule risks to the project or to life cycle support of the system, and no work-around is known.
3.
 - a.) Adversely affect the accomplishment of an operational or mission essential capability, but a work-around solution is known.
 - b.) Adversely affect technical, cost, or schedule risks to the project or to life cycle support of the system, but a work-around is known.
4.
 - a.) Results in user/operator inconvenience or annoyance, but does not affect a required operational or mission essential capability.

- b.) Results in inconvenience or annoyance for development or support personnel, but does not prevent the accomplishment of the responsibilities of these personnel.
- 5.
 - a.) This priority denotes any other effect.

Priority is a measure of the impact and elements of importance that occur during a test process. The impact and elements of importance are considered in defining the severity of a system problem. In the SI&T process, severity is defined as the degree to which a problem adversely influences system operation or the overall test effort. The effect of a problem can range from extreme (e.g., data loss or hardware damage) to minimal (e.g., cosmetic display screen error). Both the SD group and the Test group must review and agree on the level of severity. Severity is represented in terms of high, medium, or low.

During the SPR Review, each SPR is assigned to the SD group, identified as a duplicate of a previously reported software problem, or postponed until a later date for action. The SPR Review process is not limited to new submissions or SPRs previously postponed. The SPR Review must also consider if SPRs are being processed appropriately and expeditiously.

During its life cycle, a SPR will exist in one of the following states:

- Submitted
- Postponed
- Duplicate
- Assigned
- Opened
- Resolved
- Closed

The time lapsed between states is used to assess the quality of the SPR identification and resolution process. For this reason, duplicates are identified and remain in that state without being closed.

In order to facilitate the tracking process and support testing report requirements, the following data elements must be captured from each SPR form upon submission:

- A unique identifier for the SPR.
- A description of the problem.
- The specific test case and software version in which the problem was discovered.
- The date on which the problem was reported.
- The test engineer or observer who reported the problem.

As the SPR is tracked through the life cycle, more information is added to the tracking database. The following data elements are added or changed upon progression:

- The current state is modified.
- The date on which the SPR was last reviewed.
- The member of the development team to which the SPR was assigned.
- The date on which the developer opened the SPR for analysis.
- The date on which the developer changed the SPR state to resolved.
- The date on which a test engineer last tested for validation of the resolution.
- The date on which the test engineer returned the item to the “Opened” state because the item failed re-test and resolution could not be validated.
- The severity of the problem, as determined by the SPR Review process.
- The priority for the repair of the problem, as determined by the SPR Review process.

APPENDIX C

SYSTEM PROBLEM STATUS REPORT GUIDELINES

This page intentionally left blank

System Problem Report Status Report Guidelines

Field Name	Description
Testing Phase	The current phase of testing (i.e., unit, integration, performance, SQT).
Date of Report	The date the report was completed.
Date of Last Review	The date of the last SPR Review.
Software Version	Current software version.
Reported By	CM group member who completed the report.
Number of submitted SPRs itemized by Priority	The number of submitted SPRs itemized by priority.
Number of assigned SPRs itemized by Priority	The number of assigned SPRs itemized by priority.
Number of opened SPRs itemized by Priority	The number of opened SPRs itemized by priority.
Number of resolved SPRs itemized by Priority	The number of resolved SPRs itemized by priority.
Number of closed SPRs itemized by Priority	The number of closed SPRs itemized by priority.
Number of postponed SPRs itemized by Priority	The number of postponed SPRs itemized by priority.
Number of duplicated SPRs itemized by Priority	The number of duplicated SPRs by priority.
Grand total of all SPRs to date.	Total SPRs written.
Number of re-opened SPRs itemized by Priority	The number of re-opened SPRs itemized by priority.

This page intentionally left blank

APPENDIX D

TEST READINESS REVIEW REPORT GUIDELINES

This page intentionally left blank

Test Readiness Review Report Guidelines

The Test Readiness Review (TRR) is a formal review that determines readiness to begin a formal testing phase. Upon completion of formal reviews, a milestone is reached in the SI&T process, and a new phase of testing begins. After the CSCI Integration Testing is complete in the SU Test phase, an Integration Testing TRR is held. Similarly, after the Integration Testing is completed a Performance Testing TRR is held. Finally, once Performance Testing is completed, a System Qualification Testing TRR is held.

The purpose of each TRR is for the STO to determine whether the next phase of testing is ready to begin. A technical understanding is reached on the formal test results, and the validity and degree of completeness of the system testing that occurred during the preceding SI&T phase is determines.

The items to be reviewed are the same for each TRR and are identified in Figure D.1.

Item	Description
Requirements Changes	Any changes to the Software Requirements, Interface Requirements Specification(s) or other system requirement documentation approved since the last baseline STP and STD were created.
Design Changes	Changes to the Software Top-Level Design Document, Software Detailed Design Document, Data Base Design Document(s), or Interface Design Document(s) made since the last baseline STP and STD were created.
STP and STD	Changes to approved STP and/or STD made since the last baseline.
Upcoming Test Cases and Test Case Procedures	Test case(s) and test case procedure(s) to be used during conduct of the next testing phase to meet the requirements of the baseline STD.
Results of Previous Tests	Test case(s) and test case procedure(s) used during conduct of the most recent tests and the results of those tests placed under CM control.
Software Test Resources	Status of testing facility hardware, software, test personnel, and supporting test software and materials. This includes software test tool qualification and review of traceability between requirements and their associated tests are approved for the next test phase.
Test Limitations	Identification of all software test limitations.
Outstanding SPRs	Summary of SPR status including all known discrepancies of the test support software.

Item	Description
Schedules	Schedules set for remaining milestones.
Documentation Updates	Updates to all evolving and baseline system documentation.

Figure D.1 TRR Items

Appropriate members of the software test organization will present items described in Table D.1. Other members of the STO will determine the effect on their portion of the SI&T process and either approve or disapprove artifacts or information.

APPENDIX E

QUALITY ASSURANCE CHECKLISTS

This page intentionally left blank